Thomas Bennett - trb090020
CS 4395.001 - UTD Spring ‘23
N-gram Narrative

N-grams

What is an n-gram?

An n-gram is a sliding window structure of size n over some text. A unigram is an n-gram of size 1, and a bigram is a n-gram of size 2. N-grams can be used to build a model of language since they represent sequences of words in the language being modeled. By creating n-grams for a language you can classify a body of text as falling into a category of language by calculating the frequency of an n-gram appearing in the text.

Applications of N-grams

- Classifying bodies of text as belonging to a language.
- Suggestion models for predicting the next word in a sentence or statement.
    - Predicting the next word or operator in a block of code.
    - Predicting the next word in a word processor or text message application
- Detecting spelling errors
    - Sliding window over letters in a word
- Language translation
- Biological sequences
    - DNA
    - Protein
- Improved data compression ratios

Probabilities of Unigrams and Bigrams

The probability of a unigram, *u*, is calculated by dividing the number of times a distinct *u* appears in the list of all unigrams in a text by the number of total unigrams in the text.

The probability of a bigram, *b*, is calculated by dividing the number of times a distinct *b* appears in the list of all bigrams by the number of unigrams which have the same word as the first word in *b*.

Building a Language Model

If n-grams are to be used as a model of language, the corpus upon which the n-grams are trained determines the probabilities of n-grams. Therefore, a corpus to be used to create a model of language should be large, diverse, and relevant to what the model seeks to classify.. A language model will incorporate the biases present in the corpus used to create it. For example,

using a corpus consisting of only English sentences to create a language model would be a very poor choice for a model designed to classify text into classes of languages.

Smoothing

It is not the case that all possible n-grams will always exist in the n-gram model. Therefore, it is possible that the probability of a certain n-gram is 0. Since we multiply the probabilities together to produce a probability that some text is a member of some class of language, a 0 anywhere in the chain of probabilities will cause the calculated probability to be zero. Smoothing is the act of giving a very small value to probabilities that would otherwise be 0 to avoid the problem of having a 0 in the chain of probabilities. A simple approach to smoothing is to add a 1 to the numerator of all probability calculations.

Text Generation using N-grams

N-grams can be used to generate text by using bigrams, or greater sized n-grams, to make a prediction about what the next word should be. The initial word can be chosen randomly, and then the highest probability n-gram which begins with that word is chosen for generating the next word or words of the sentence.
The drawback to this is that the generation is based entirely on probability, which is not how human diction works, so the generated text may be unintelligible.

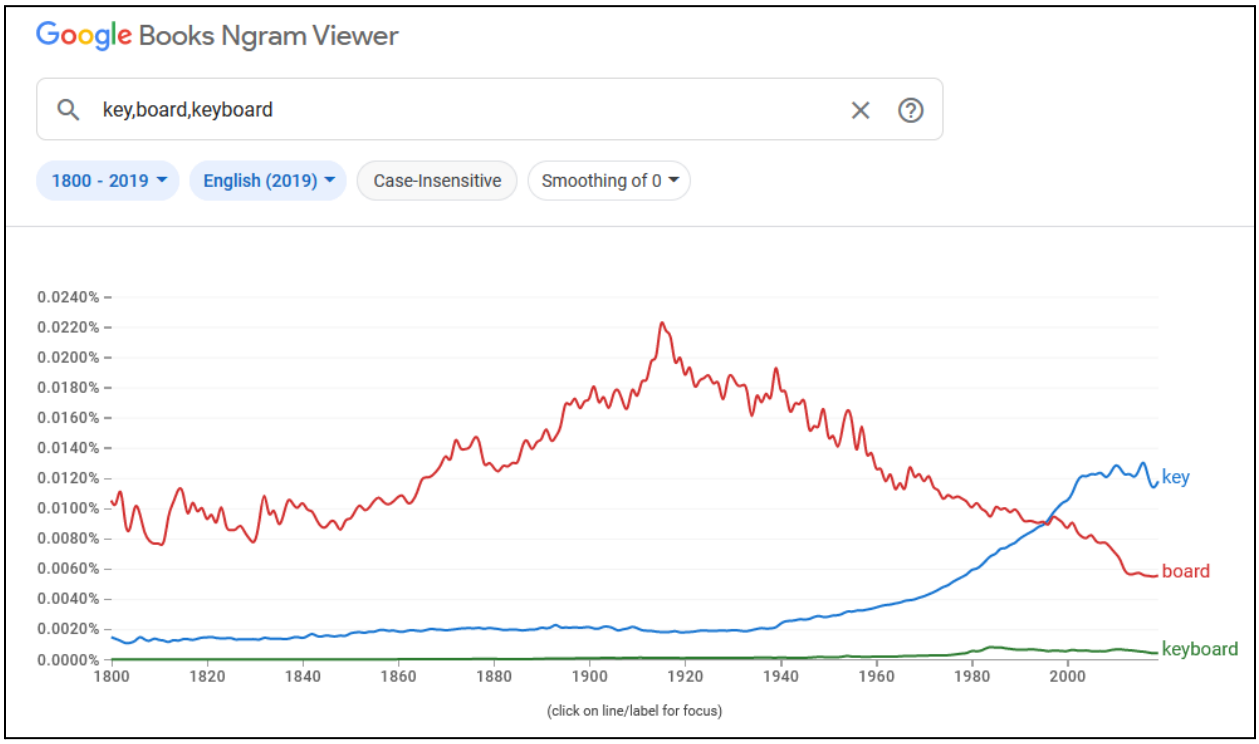
Evaluating Language Models

A language model can be evaluated by its classification performance against known results of classifying language. However, annotated solutions require human actors which are slow and expensive.
Perplexity, the inverse probability of seeing the words observed, normalized by the number of words, is an intrinsic evaluation metric.

Google’s N-gram Viewer

Google has a publicly-available n-gram viewer which was created using the Google Books corpus. A user may enter n-grams, separated by commas, and the application displays a chart of the historical frequency of the n-grams, from the 1800 to the year 2019, by default.

Thomas Bennett - trb090020
CS 4395.001 - UTD Spring ‘23
N-gram Narrative



Sample run: key, board, keyboard